Feature selection for an SVM based webpage classifier

Mtetwa, Nhamoinesu; Yousefi, Mohammadmehdi; Reddy, Viseshini

Published in:
IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI), 2017

DOI:
10.1109/ISCMI.2017.8279603

Publication date:
2018

Document Version
Author accepted manuscript

Feature selection for an SVM based Webpage classifier

Nhamo Mtetwa, Mehdi Yousefi,
CCIS Department, Glasgow Caledonian University
Glasgow, United Kingdom
E-mail: {nhamoinesu.mtetwa, mehdi.yousefi, }@gcu.ac.uk

Vineshini Reddy
Department of Computer Science and Engineering
Manipal University
Manipal, India
Viseshini.reddy@gmail.com

***Abstract*—Machine-learning techniques are a handy tool for deriving insights from data extracted from the web. Because of the structure of web data extracted by web crawlers there is need for preprocessing the data to extract features that can be used to train a machine learning classifier. The number of available features that can be linked to a website is huge. Narrowing down to a minimum number of features required to drive a classifier has huge benefits. This paper presents a workflow that uses a set of metrics that can be used to reduce the numbers of features for training a support vector machine (SVM) for classifying webpages as fraudulent or not. The paper reports that a three quarter reduction in feature set size only incurs a 5% reduction in classification accuracy which has huge computational benefits.**

Keywords-feature extraction; machine learning; web crawling; information; support vector machine

I. INTRODUCTION

We live in a digitally connected world and a major part of it is the World Wide Web which is growing at an exponential rate [1]. The web has become the go to place for almost everything because of the vast amounts of information it contains. Unfortunately, the web is used for both good and bad which makes the ability to classify webpages as either good or bad important. Because of the ever-growing size of the web there is need for continual improvement in the tools for classifying webpages at scale. Machine learning is one such tool that is used to classify webpages [2]. A fundamental part of machine learning is to approximate the functional relationship $f(.)$ between an input vector X and an output Y [3]. Sometimes the output Y is not determined by the complete set of the input features, instead, it is decided only by a subset of them. With sufficient data and time, it is fine to use all the input features, including those irrelevant features, to approximate the underlying function between the input and the output. But in practice, there are two problems that may be introduced by the irrelevant features involved in the learning process. First, the irrelevant input features may result in a greater computational cost and second, the irrelevant input features may lead to over fitting. In this paper three metrics for measuring how much each feature contributes to the classification of webpages are discussed. The metrics are information gain [4], principal components and chi-squared [5].

Extracting data from the web is an important problem that has been tackled using different tools and in a broad range of applications [6]. At the Enterprise level, web data extraction techniques emerge as a key tool to perform data analysis in business and competitive intelligence systems as well as for business process re-engineering. At the social web level, web data extraction techniques allow businesses to gather a large amount of data continuously generated and disseminated by Web 2.0, Social Media and Online Social Network users and this offers unprecedented opportunities to analyse human behaviour at a very large scale.

There are two main ways of extracting data from a website: one can either use APIs which the website exposes for other applications to easily extract data from it or one could use web crawlers or spiders, a technology used by search engines. The crawler downloads the html files which constitute the website. In this work a web crawler is used to download html files, another application is then used to extract feature data from the html file to analyse the files and a third application is used to evaluate the features for the classification task. Finally, a support vector machine based classifier is trained to classify the web pages.

The rest of the paper is organized as follows: in section II, we review some work on feature selection. In section III, the details of the methodology for the practical implementation are discussed. In section IV, we present the results. Finally, in section V, the paper is concluded.

II. LITERATURE REVIEW

There is a lot of focus on machine learning but not enough focus on what happens before the machine-learning step. Machine learning models, such as neural networks, decision trees, random forests and gradient boosting machines accept a feature vector and provide a prediction [7]. These models learn in a supervised fashion where a set of feature vectors with expected output is provided [8]. Feature selection has become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available [9]. These areas include text processing of Internet documents, gene expression array analysis, and combinatorial chemistry [10]. Feature selection is the process of selecting a subset of relevant features (variables, predictors). The objective of feature selection is three-fold: improving the prediction

performance of the predictors/classifiers, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data [11]. In this paper, feature selection for use in a webpages classification model construction is discussed. The paper discusses three different metrics for selecting the best set of features from semi-structured webpages data. The target application for this feature selection exercise is classification of webpages as either fake or not.

Many approaches to extracting data from the Web have been designed to solve specific problems and operate in ad-hoc domains [6]. Other approaches, instead, heavily reuse techniques and algorithms developed in the field of Information Extraction. Various machine-learning applications are usually overwhelmed by a large number of features.

A *feature* is a numeric representation of raw data [12]. There are many ways to turn raw data into numeric measurements. Basically, features must derive from the type of data that is available. Perhaps less obvious is the fact that they are also tied to the model; some models are more appropriate for some types of features, and vice versa. The right features are relevant to the task at hand and should be easy for the model to ingest. *Feature selection* is the process of formulating the most appropriate features given the data, the model, and the task [11]. The NIPS 2003 Feature Selection Challenge offered a great testbed for evaluating feature selection algorithms on datasets with a very large number of features as well as relatively few training examples [13].

The number of available features that can be linked to a webpage or an email is huge. These features are associated with certain website's elements such as the URL, domain, and source code. One primary challenge in minimizing the website fraud risk is to identify the smallest set of features before classifying the website as fraudulent or legitimate. Not considering this challenge may cause deterioration in the fraud detection rate especially when many redundant features are kept in the dataset. These redundant features increase the search space for the classification algorithm.

A. *Information Gain*

Information gain (IG) measures the amount of information in bits about the class prediction, if the only information available is the presence of a feature and the corresponding class distribution [14]. It measures the expected reduction in entropy (uncertainty associated with a random feature) [15].

B. *Principal Components and Analysis (PCA)*

Principal component analysis (PCA) is a classical statistical method [16]. PCA is arguably the most widely used statistical tool for data analysis and dimensionality reduction today. Large datasets are increasingly common and are often difficult to interpret. PCA reduces the dimensionality of such datasets, increasing interpretability but at the same time minimizing information loss. It does so by creating new uncorrelated variables that successively maximize variance [17]. What PCA does is to discover new variables called principal components that account for the majority of the variability in the data [18]. This enables one to describe the information with considerably fewer variables or features in our case [19].

C. *Chi-Squared*

Chi-squared (CHI) is another widely used metric in machine learning for evaluating the goodness of an attribute [20]. The CHI measures the degree of independence between a pair of categorical variables [21]. In the present context, the greater the CHI score of a feature, the more independent that feature is from the class variable.

D. *Support Vector Machines*

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1. To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function.

III. METHOD

The methodology followed in this work is summarised in Fig. 1.

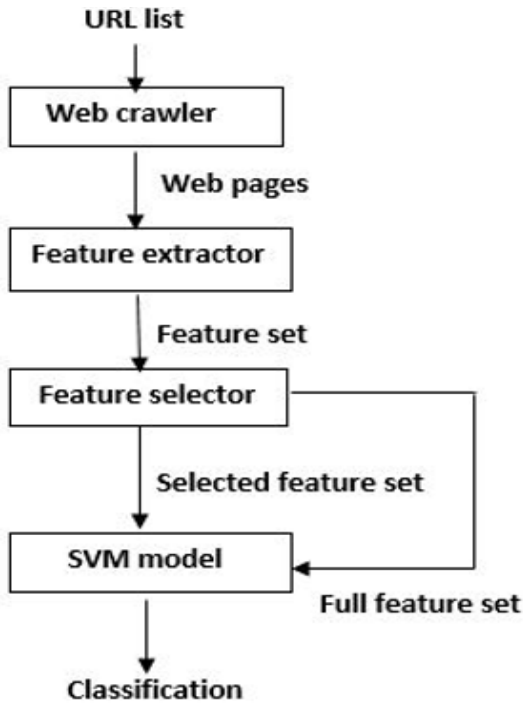

Fig. 1. Flowchart of the methodology followed

A. Data Collection

A set of URLs associated with fake websites from Artists Against 419 [22]– a database of fake sites urls was compiled together with urls from related genuine or legitimate websites. A Python program for extracting data from these websites based on Scrapy, an open source web crawler was developed [23]. A set of features associated with fake or fraudulent websites was compiled based on the work of [24]. In their work, [24] categorised the feature list into web page text, urls, source code, images and linkages. Scrapy was used to download the crawled webpages and a Python program was used to parse the files and extract features from each page's file. Each web page resulted in a feature set with an associated label of fake or legit depending on whether the page belonged to a fake or legit site. A feature database based on crawling 1000 webpages was created. The list of features can be seen in Table 1.

TABLE 1: LIST OF FEATURES USED

Features	Explanation
#AbsoluteLinks	Number of absolute links per page
#RelativeLinks	Number of relative links per page
#HTTPSLinks	Number of secure links per page
#InLinks	Number of links, in a page, pointing to the same domain as the initial URL
#OutLinks	Number of links, in a page, pointing to a different domain from the initial URL
AvgSlashes	Average number of forward slashes in each URL in a page
#PreloadedImages	Number of preloaded images in a page
#MisSpeltWords	Number of incorrectly spelt words in a page
AvgSentenceLen	Average number of words in a sentence in a page
#BadGrammar	Number of grammatically incorrect sentences in a page
AvgWordLen	Average length of the words in a page
#WordsPerPage	Number of words in a page

B. Feature Selection

A Python program was used to compute principal components, information gain and chi-squared values for all the features. The features were then ranked according to each of the above metrics. The top 3 features for each metric were used to train a support vector machine and the results were compared with a similar SVM model based on the full feature set. Python's Sci-Kit-learn machine learning library was used to train and test the SVM model. The set of metrics for selecting features was informed by the work of [24].

IV. RESULTS

The results below were based on 1000 webpages worth of data based on the features in Table 1.Table 2 shows the ranking of these features using the three metrics (information gain, PCA and Chi Square).

TABLE 2: FEATURES RANKED BASED ON THE THREE METRICS

Rank	Information Gain	PCA	Chi Square
1	AvgSentenceLen	#MisSpeltWords	MisSpeltWords
2	#MisSpeltWords	WordsPerPage	WordsPerPage
3	#AbsoluteLinks	AvgSentenceLen	AvgSentenceLen
4	AvgSlashes	#HTTPSLinks	AbsoluteLinks
5	AvgWordLen	#PreloadedImages	#InLinks
6	#InLinks	#Grammar	Grammar
7	#WordsPerPage	#AbsoluteLinks	#PreloadedImages
8	#HTTPSLinks	AvgWordLen	#HTTPSLinks
9	#PreloadedImages	#OutLinks	#OutLinks
10	#Grammar	#RelativeLinks	#RelativeLinks
11	#OutLinks	AvgSlashes	AvgWordLen
12	#RelativeLinks	#InLinks	AvgSlashes

The top 3 features for each feature selection metric were used to train an SVM model and the results are given in Table 3.

TABLE 3: SVM ACCURACY BASED ON TOP 3 FEATURES FOR EACH METRIC

Feature set	SVM Accuracy
All 12 features	82%
Top 3 Information Gain	72%
Top 3 Chi Squared features	77%
Top 3 PCA features	77%

Table 3 shows that PCA and Chi-Squared are both ranked number 1 followed by information gain in terms of classification accuracy on the SVM model. The interesting thing is that using just a quarter of the features results in a 5% loss in accuracy but a massive gain in computational efficiency. Whether a 5% loss of accuracy is a good or bad tradeoff depends on the use case. These results show that it is worth evaluating the usefulness of each feature before including it in an SVM model.

V. CONCLUSION

The number of features in a model is important. If there are not enough informative features, then the model will be unable to fulfil its ultimate task. If there are too many features, or if most of them are irrelevant, then the model could go awry in the training process which impacts the model's performance [25].

Features and models sit between raw data and the desired insight. In a machine learning workflow, we pick not only the model, but also the features. This is a double-jointed lever, and the choice of one affects the other. Good features make the subsequent modelling step easy and the resulting model more capable of achieving the desired task. Bad features may require a much more complicated model to achieve the same level of performance. The more thoughtful input features one has, the better the accuracy and efficiency of the model.

In future we will expand the feature list to include meta data about the domain names of websites like the age of the domain name, whether it is secure or not and other whois database attributes. Some of these features may have more classifying power than the features used in this paper. The goal is to increase the classification accuracy.

ACKNOWLEDGMENT

The authors would like to thank the British Council for sponsoring Viseshini Reddy's internship through the IAESTE (International Association for the Exchange of Students of Technical Experience) programme.

REFERENCES

[1] B. A. Huberman and L. A. Adamic, "Internet: growth dynamics of the world-wide web," *Nature,* vol. 401, p. 131, 1999.

[2] R. Schutt and C. OŃeil, Doing data science: Straight talk from the frontline, " OŔeilly Media, Inc.", 2013.

[3] J. Weston, A. Elisseeff, B. Schölkopf and M. Tipping, "Use of the zero-norm with linear models and kernel methods," *Journal of machine learning research,* vol. 3, pp. 1439-1461, 2003.

[4] J. T. Kent, "Information gain and a general measure of correlation," *Biometrika,* vol. 70, pp. 163-173, 1983.

[5] S. Paul, M. Magdon-Ismail and P. Drineas, "Feature selection for linear SVM with provable guarantees," *Pattern Recognition,* vol. 60, pp. 205-214, 2016.

[6] E. Ferrara, P. De Meo, G. Fiumara and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowledge-based systems,* vol. 70, pp. 301-323, 2014.

[7] Y. Saeys, T. Abeel and Y. V. de Peer, "Towards robust feature selection techniques," in *Proceedings of Benelearn*, 2008.

[8] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon, 2016*, 2016.

[9] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the ninth international workshop on Machine learning*, 1992.

[10] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science,* vol. 91, pp. 919-926, 2016.

[11] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, pp. 1157-1182, 2003.

[12] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers \& Electrical Engineering,* vol. 40, pp. 16-28, 2014.

[13] I. Guyon, S. Gunn, A. Ben-Hur and G. Dror, "Result analysis of the NIPS 2003 feature selection challenge," in *Advances in neural information processing systems*, 2005.

[14] D. Roobaert, "Information gain, correlation and support vector machines.," 2006.

[15] T. M. Mitchell, "Machine learning and data mining," *Communications of the ACM,* vol. 42, pp. 30-36, 1999.

[16] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, MIT press, 2016.

[17] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Phil. Trans. R. Soc. A,* vol. 374, p. 20150202, 2016.

[18] I. T. Jolliffe, "Discarding variables in a principal component analysis. I: Artificial data," *Applied statistics,* pp. 160-173, 1972.

[19] A. Davies, *Sepctroscopy,* pp. 20-23, June 2016.

[20] K. D. Rajab, "New Hybrid Features Selection Method: A Case Study on Websites Phishing," *Security and Communication Networks,* vol. 2017, 2017.

[21] F. Provost and T. Fawcett, Data Science for Business: What you need to know about data mining and data-analytic thinking, " OŔeilly Media, Inc.", 2013.

[22] *https://db.aa419.org/fakebankslist.php.Web page accessed on July 13, 2017.*

[23] *https://scrapy.org. Web page accessed on July 14, 2017.*

[24] A. Abbasi, Z. Zhang, D. Zimbra, H. Chen and N. a. J. F. Jr, "Detecting fake websites: the contribution of statistical learning theory," *Mis Quarterly,* pp. 435-461, 2010.

[25] O. Chapelle and S. S. Keerthi, "Multi-class feature selection with support vector machines," in *Proceedings of the American statistical association*, 2008.